REMARKS

Introduction

Claims 1-37 are pending in this application.

Applicant has amended claims 1, 20, 24, 26, 27, 34, and 37 to more particularly define the invention and to correct minor clerical and/or typographical errors. No new matter has been added and the amendments are fully supported and justified by the specification.

Reconsideration of this application in light of the following remarks is hereby respectfully requested.

Summary of the Office Action

Claims 27 and 37 are rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

Claims 20 and 37 are objected to for containing informalities and for using indefinite language.

Claims 1, 2, 4, 5, 7, 8, 24-26, and 34-36 are rejected under 35 U.S.C. § 102(e) as being anticipated by Panchul et al., U.S. Patent No. 6,226,776 (hereinafter "Panchul").

Claim 20 is rejected under 35 U.S.C. § 102(e) as being anticipated by Killian et al., U.S. Patent No. 6,477,683 (hereinafter "Killian").

Claims 3, 27, and 37 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Panchul.

Claims 6, 9-12, 16-19, and 28-33 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Panchul in view of Killian.

Claims 13-15, and 21-23 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Panchul in view of Killian in view of Ashar et al., U.S. Patent No. 6,745,160 (hereinafter "Asher").

Applicant's Reply to the Rejection of
Claims 27 and 37 under 35 U.S.C. § 101

The Examiner rejected claims 27 and 37 under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Although applicant does not agree with Examiner's basis for rejection, applicant has amended independent claims 27 and 37 to further advance prosecution of this application.

Applicant's amended claim 27 is directed toward a method for mapping software constructs into hardware constructs based on the software constructs. The software constructs are parsed and software construct variables are mapped into a hardware construct. The hardware construct corresponding to the software construct variable contains a

set of wires of which one indicates whether the variable has been computed. The remainder of the wires indicate the value of the variable. Applicant's amended claim 37 is directed toward a hardware construct implemented in programmable logic in which a software construct variable is implemented in hardware as a set of wires (e.g., as recited in claim 22).

The Examiner asserts that claims 27 and 37 are directed to non-statutory subject matter. In particular the Examiner asserts that claims 27 and 37 "fail[] to recite any concrete action as to further specify how the mapping as claimed is being implemented." Office Action, page 2. The Examiner further asserts that "the claim[s] fail[] to provide a concrete and tangible result as required by the practical application test; hence amounts to an abstract idea." Id.

Contrary to the Examiner's assertions, the Supreme Court has held that Congress chose the expansive language of 35 U.S.C. § 101 so as to include "anything under the sun that is made by man." Diamond v. Chakrabarty, 447 U.S. 303, 309 (1980). Applicant's original claims 27 and 37 recited that the mapping of software constructs to hardware constructs is implemented using a set of wires in hardware to represent a software variable. More particularly, one of the wires indicates whether the variable has been computed whilst the

remaining wires indicate the value of the variable. Accordingly, applicant respectfully submits that the claims recite concrete actions and that the concrete and tangible result of a hardware construct containing a set of wires representing a software variable is not an abstract idea.

However, in response to the Examiner's rejections, applicant has amended claims 27 and 37 in an effort to advance prosecution. In particular, applicant has amended claim 27 to clearly recite the actions taking place and the concrete results.

Accordingly, applicant respectfully requests that rejection of claims 27 and 37 under 35 U.S.C. § 101 be withdrawn.

Applicant's Reply to the
Objections to Claims 20 and 37

The Examiner objected to claim 20 because of an informality in the claim. Applicant has amended claim 20 to correct this alleged informality. Accordingly, applicant respectfully requests that the objection to claim 20 be withdrawn.

The Examiner objected to claim 37 because of indefinite language in the claim. Applicant has amended claim 37 to remove the alleged indefinite language.

Accordingly, applicant respectfully requests that the objection to claim 37 be withdrawn.

Applicant's Reply to the Rejection of
Claims 1, 24, and 34 under 35 U.S.C. § 102(e)

The Examiner rejected claim 1, 24, and 34 under 35 U.S.C. § 102(e) as being anticipated by Panchul. The Examiner's rejections are respectfully traversed.

Applicant's invention, as defined by amended independent claim 1, is directed toward a method for generating hardware configuration data from software constructs. High-level software programming code, transparent with regard to hardware resources and hardware configuration, is parsed and hardware configuration data is compiled directly from the high-level software programming code.

Applicant's invention, as defined by amended independent claims 24 and 34, is directed toward a method and a programmable logic resource for directly mapping software constructs of a program into hardware constructs.

Panchul refers to a computer aided hardware design system for converting a high-level programming language, such as ANSI C, into a register transfer level hardware description language (HDL). The high-level programming

language, once converted into HDL, can be simulated and synthesized into a gate-level hardware representation.

The Examiner asserts that Panchul shows all of the elements of applicant's independent claims 1, 24, and 34. Applicant respectfully submits that Panchul fails to show or suggest the elements of applicant's amended independent claims. In particular, Panchul refers to a system for compiling a high-level programming language into an HDL which may be further synthesized into a hardware representation of the HDL. Thus, Panchul shows a system for converting a high-level programming code, which is transparent with regard to hardware resources, into an HDL code. The HDL code, which is not transparent with regard to hardware resources, may in turn, be compiled into hardware configuration data. In contrast, applicant's invention permits high-level software programming code to be directly compiled into hardware configuration data without the extraneous steps of generating HDL code. Panchul fails to make any mention of generating hardware configuration data directly from high-level software code having no hardware descriptors.

Accordingly, for at least the above reasons, applicant's amended independent claims 1, 24, and 34 are

allowable over Panchul. Applicant respectfully requests that the rejection of claims 1, 24, and 34 be withdrawn.

Applicant's Reply to the Rejection of Claim 20 under 35 U.S.C. § 102(e)

The Examiner rejected claim 20 under 35 U.S.C. § 102(e) as being anticipated by Killian. The Examiner's rejection is respectfully traversed.

Applicant's invention, as defined by amended claim 20, is directed toward a method for optimizing hardware that is generated by a software-to-hardware compiler. The software-to-hardware compiler locates at least one expression in a software program that is used more than once. The compiler then uses a single set of hardware resources to implement the multiple instances of the software expression. Run-time decisions are then made whether to use the hardware resources for each instance of the software expression. For example, a repeatedly used software expression may be implemented using a single shared hardware block (i.e., instead of having a separate hardware block for each instance of the software expression). At run-time an arbitrator, for example, can control access to the shared hardware block.

Killian refers to an automated processor design tool. The tool uses a description of a customized processor

instruction set to develop an HDL description of the processor. While compiling the processor, the processor's operation can be optimized by searching a program code for multiple instruction patterns that are repeated in the code and replacing the multiple instruction patterns with a single custom instruction. Killian, column 19, lines 35-49.

The Examiner asserts that Killian shows applicant's feature of "selecting at runtime instances that will have access to the single set of hardware resources." The Examiner further asserts that "partitioning the hardware in blocks by a HDL specification and compilation inherently yields runtime instances that will access a set of hardware resources." Office Action, pages 5-6. Applicant respectfully disagrees with the Examiner's assertions.

When a reference is silent about an asserted inherent characteristic, evidence is required to show that the missing descriptive matter is necessarily present. See Continental Can Co. USA v. Monsanto Co., 948 F.2d 1264, 1268 (Fed. Cir. 1991). The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. See In re Rijckaert, 9 F.3d 1531, 1534 (Fed. Cir. 1993). Applicant respectfully submits that the Examiner

has failed to provide any evidence with respect to the assertion that synthesizing hardware from an HDL code inherently shows applicant's claimed feature of selecting at run-time instances that will access a set of hardware resources.

Notwithstanding this unsupported assumption by the Examiner, applicant respectfully submits that Killian cannot show applicant's claimed feature, either expressly or inherently, of selecting at run-time instances that will have access to the single set of hardware resources. In particular, the partitioning of hardware referred to by Killian is performed during compilation and not during run-time. Thus, Kilian does not show hardware instances selected at run-time as required by applicant's independent claim.

Accordingly, for at least the above reasons, applicant's amended independent claim 20 is allowable over Killian. Applicant respectfully requests that the rejection of claim 20 be withdrawn.

Applicant's Reply to the Rejection of
Claims 27 and 37 under 35 U.S.C. § 103(a)

The Examiner rejected claims 27 and 37 under 35 U.S.C. § 103(a) as being obvious over Panchul. The Examiner's rejections are respectfully traversed.

The Examiner asserts that applicant's independent claims 27 and 37 are obvious over Panchul. The Examiner concedes that "Panchu[l] does not disclose data wires and a computed wire to denote the variable is valid." Office Action, page 6. However, the Examiner takes official notice that "a set of data being passed through a memory port or a circuit gate and such passage is being enabled by a valid or enable Bit was a known concept in the art." Id. Applicant respectfully disagrees with the Examiner's assertion.

As the Examiner conceded, Panchul does not show applicant's claimed invention. Nor does the combination of Panchul with the Examiner's official notice (assuming it is a valid official notice) show applicant's claims invention. Panchul refers to mapping software variables into hardware registers and hardware memory. Panchul, column 5, lines 46-55. In contrast, applicant's invention implements software variables as a set of wires in order to avoid mapping software variables into hardware registers. For example, as applicant's specification clearly states, "this implementation [of software variables as registers] inherently leads to inefficiencies," and "[i]n order to avoid the use of too many large multiplexers, variables may be implemented in hardware as a set of wires." Applicant's

Specification, page 2, lines 23-25, and page 4, lines 8-10. Panchul fails to show or suggest such an improvement over the reliance on registers.

Further, there is no motivation to modify Panchul in the manner suggested by the Examiner. The Examiner asserts that the motivation for this combination comes from the fact that wires transporting data require control and that a control bit or wire would provide this control. Office Action, page 7. However, there are many techniques that may be used to provide this control. Applicant respectfully submits that the Examiner has not provided an objective teaching that would suggest this modification. Rather, applicant submits that with the knowledge of applicant's novel method for representing a software construct variable as a hardware construct, particular techniques in the prior art were identified for use in rejecting applicant's invention. This process has long been held invalid by the courts at creating a *prima facie* case of obviousness. See In re Fine, 5 USPQ2d at 1600. ("One cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention.").

Without a proper motivation for combining the references, the Examiner has "simply take[n] the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability," a practice that is insufficient as a matter of law. In re Dembiczak, 175 F.3d 994, 999. In doing so, the Examiner has demonstrated mere hindsight reconstruction, the very "syndrome" that the requirement for objective evidence is designed to combat, and the rejection is therefore insufficient as a matter of law. Id.

Accordingly, for at least the above reasons, applicant's amended independent claims 27 and 37 are allowable over Panchul. Applicant respectfully requests that the rejection of claims 27 and 37 be withdrawn.

Applicant's Reply to the Rejection of
Claims 9 and 28 under 35 U.S.C. § 103(a)

The Examiner rejected claims 9 and 28 under 35 U.S.C. § 103(a) as being obvious over Panchul in view of Killian. The Examiner's rejections are respectfully traversed.

Applicant's invention, as defined by independent claims 9 and 28, is directed toward a method and programmable logic resource for exploiting parallelism by making speculation decisions at run-time. A control flow is

generated in the hardware, in which the control flow indicates the status of a block (e.g., status reflects the block's capability for speculation). At run-time, the hardware can make decisions regarding execution of the block at least partially based on the control flow. For example, hardware control flow is synthesized using a special set of control flow wires that are used to enable or disable operations within blocks. One of the operations enabled by the control flow wires is speculative execution (i.e., the ability to execute a block even though it is not guaranteed that the result will be needed). The control flow values of a block may change during execution to allow speculation.

The Examiner asserts that Panchul shows a method for generating hardware that exploits parallelism by making decisions at run-time. The Examiner concedes that Panchul does not disclose having a status indicative of a block's capability for speculation. However, the Examiner asserts that Killian shows "provid[ing] speculation to the implementation from compiling HDL into hardware." Office Action, page 8. The Examiner also asserts that the concept of "a control bit, a variable bit, or a predicate check in order to establish whether a chunk of instructions can be executed speculatively at compilation was a known concept in

the art of compiler optimization at the time of the invention." Id. The Examiner concludes that the combination of Panchul, Killian, and "the art of compiler optimization" show all of the elements of applicant's independent claim. Applicant respectfully disagrees with the Examiner's assertions.

Panchul does not refer to a method for generating hardware that exploits parallelism by making decisions at run-time. In particular, the Examiner describes the system of Panchul as "making determination[s] to exploit parallel execution based on HDL compilation and configuration." Id. Any decisions about parallelism made by the system of Panchul occur during compilation into HDL and not at run-time. Thus, Panchul does not show all of the elements of applicant's independent claims. Killian also fails to show at least this feature of applicant's claimed invention.

Moreover, there is no motivation to modify Panchul in the manner suggested by the Examiner. The Examiner asserts that it would have been obvious to modify Panchul because "speculative executions can avert data fetching exceptions." Id. Applicant respectfully submit that the Examiner has only alluded to a general benefit of speculation without providing an objective teaching that would provide

motivation to modify Panchul to generate hardware that exploits parallelism by making decisions at run-time, as required by applicant's amended claim.

Applicant further submits that with the knowledge of applicant's novel invention, particular techniques in the prior art were identified for use in rejecting applicant's invention. This practice has long been held invalid by the courts at creating a *prima facie* case of obviousness. See In re Fine, at 1600. ("One cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention.").

Without a proper motivation for combining the references, the Examiner has "simply take[n] the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability," a practice that is insufficient as a matter of law. In re Dembiczak at 999. In doing so, the Examiner has demonstrated mere hindsight reconstruction, the very "syndrome" that the requirement for objective evidence is designed to combat, and the rejection is therefore insufficient as a matter of law. Id.

Accordingly, for at least the above reasons, applicant's amended independent claims 9 and 28 are allowable

over Panchul. Applicant respectfully requests that the rejection of claims 9 and 28 be withdrawn.

Applicant's Reply to the Rejection of
Claims 2-8, 10-19, 21-23, 25, 26, 29-33, 35, 36

The Examiner rejected claims 2, 4, 5, 7, and 8 under 35 U.S.C. § 102(e) as being anticipated by Panchul. The Examiner rejected claim 3 under 35 U.S.C. § 103(a) as being unpatentable over Panchul. The Examiner rejected claim 6 under 35 U.S.C. § 103(a) as being unpatentable over Panchul in view of Killian. Claims 2-8 are allowable at least because they depend from allowable independent claim 1. Applicant respectfully requests that the rejection of claims 2-8 be withdrawn.

The Examiner rejected claims 25, 26, 35, and 36 under 35 U.S.C. § 102(e) as being anticipated by Panchul. Claims 25, 26, 35, and 36 are allowable at least because they depend from allowable independent claims 24 and 34. Applicant respectfully requests that the rejection of claims 25, 26, 35, and 36 be withdrawn.

The Examiner rejected claims 21-23 under 35 U.S.C. § 103(a) as being unpatentable over Panchul in view of Killian in view of Ashar. Claims 21-23 are allowable at least because they depend from allowable independent claim

20. Applicant respectfully requests that the rejection of claims 21-23 be withdrawn.

The Examiner rejected claims 10-19 and 29-33 under 35 U.S.C. § 103(a) as being unpatentable over Panchul in view of Killian. Claims 10-19 and 29-33 are allowable at least because they depend from allowable independent claims 9 and 28. Applicant respectfully requests that the rejection of claims 10-19 and 29-33 be withdrawn.

Conclusion

For at least the foregoing reasons, applicant respectfully submits that this application is in condition for allowance.

Accordingly, prompt reconsideration and allowance of this application are respectfully requested.

Respectfully submitted,

Michael J. Chasan
Registration No. 54,026
Agent for Applicant
FISH & NEAVE LLP
Customer No. 36981
1251 Avenue of the Americas
New York, New York 10020-1105
(212) 596-9000